

KeyFrame Extraction for Human Motion Capture Data via Multiple Binomial Fitting

Chenxu Xu¹, Wenjie Yu¹, Yanran Li⁴, Xuequan Lu⁵, Meili Wang^{*123} and Xiaosong Yang⁴

¹College of Information Engineering, Northwest A&F University, Yangling 712100, Chian

²Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture, Yangling 712100, China

³Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service, Yangling 712100, China

⁴Bournemouth University, NCCA

⁵Deakin University - Waurn Ponds Campus, School of Information Technology

Abstract

In this paper, we make two contributions. The first is to propose a new keyframe extraction algorithm, which reduces the keyframe redundancy and reduces the motion sequence reconstruction error. Secondly, a new motion sequence reconstruction method is proposed, which further reduces the error of motion sequence reconstruction. Specifically, we treated the input motion sequence as curves, then the binomial fitting was extended to obtain the points where the slope changes dramatically in the vicinity. Then we took these points as inputs to obtain keyframes by density clustering. Finally, the motion curves were segmented by keyframes and the segmented curves were fitted by binomial formula again to obtain the binomial parameters for motion reconstruction. Experiments show that our methods outperform existing techniques, in terms of reconstruction error.

Keywords: keyframe extraction, motion capture, curve simplification, computer animation

1 Introduction

Motion capture plays an important role in computer games, virtual environments, movie effects,

robotics [1, 2, 3], motion prediction [4], and other fields. Motion capture originates from authentic motion, so even complex motion can produce real effects in real time, which is very attractive for related fields. However, there still exist significant problems while using motion capture data. Human motion data is usually captured at high frequencies, which leads to a very large amount of data. As a result, it is difficult to the storage, retrieval, browse and reuse the capture data. It seems that keyframe extraction brings hope for solving the problems. As its name suggests, the technique of keyframe extraction is to select the most representative frames from the motion sequence to represent the whole motion sequence. It eliminates redundant frames in motion sequences, saves the space needed for storage, facilitates the browsing and editing of motion sequences, and improves the reuse efficiency of motion capture sequences. From what has been mentioned above, keyframes should have characteristics as follows: (1) Keyframes should be as few as possible while reflecting the overall movement trend of motion sequence. (2) Keyframes can reconstruct the motion sequence with minimum error by interpolation.

In this paper, a curve fitting method is proposed to extract keyframes from motion capture sequences. First, the input motion data is regarded as a set of rotation information curves,

*Corresponding author: wml@nwsuaf.edu.cn

because except for three records that are the locus information of the root joint, all the other records are the rotation information of the joint. Then identify the areas of the curve where the slope changes dramatically and the points in these areas can best represent the overall trend of joints motion changes. We employ binomial to piecewise fit the rotation information curves, make the curves fit to the specified goodness-of-fit R^2 , and get the segment points. Then we cluster the segment points, and the obtained clustering centers are optimized to obtain the keyframes. At the same time, we also propose a method of motion data reconstruction based on several areas binomial fitting. Its reconstruction error is lower than the current popular method.

2 Related Work

2.1 KeyFrame Extraction

The methods of keyframe extraction mainly include curve simplification-based, matrix decomposition-based, cluster-based and optimization-based. Curve simplification takes a frame as a point and a motion sequence as a curve in a high-dimensional space, and realizes keyframe extraction by finding a series of points that can depict the whole curve well[5, 6, 7]. However, the keyframe is extracted according to the local extremum of the motion curve, and other parts of the curve with obvious slope change are ignored. The method based on matrix decomposition is to represent motion sequence as motion matrix, and then decompose the matrix into weight matrix and keyframe matrix approximately[8, 9], but this method is time-consuming and easy to ignore the time information, and the extracted keyframes may not cover the whole sequence.

Algorithms concerning clustering-based and optimization-based keyframe extraction have become hot topic in recent years[10, 11]. B. Sun et al.[12] proposed a keyframe extraction method based on affine propagation (AP) clustering algorithm to adaptively search for the best keyframe of video. The algorithm is fast and can extract high-quality keyframes, which is also easy to implement. Q. Zhang et al.[13] used ISODATA dynamic clustering algorithm to cluster all frames and extracted the frames close to the clustering center as keyframes. This method is suitable for

a variety of motion types without user-specified parameters. However, the keyframes extracted by clustering method are not in chronological order, and it is difficult to represent the original motion sequence information.

Y. Zhang et al.[14] proposed a grey Wolf optimization algorithm, in which the extracted keyframes can well summarize the original motion sequence and maintain the consistency of keyframes between similar motion sequences. A genetic algorithm based on multiple populations proposed by Q. Zhang[15] et al. mimics natural selection and evolution mechanism, and the evolution process does not need to artificially specify threshold parameters, with fast convergence speed, but still cannot specify the number of keyframes. G. Y. Xia et al.[16] proposed a novel model called "Joint kernel sparse Representation", which could perfectly simulate the sparseness of human motion and Riemannian manifold structure, and extract keyframes from it. However, it does not effectively solve the problem of keyframe redundancy, and the time complexity is $O(n^2)$.

2.2 Motion data reconstruction

Linear interpolation and quaternion interpolation are commonly used in motion reconstruction [5, 6, 12, 14]. In animation, linear interpolation and quaternion interpolation are very good for intermediate frame compensation. Using the data fitting method to correct the motion capture error caused by deleting force feedback can effectively reduce the traditional motion error and has good application value[17]. However, if the original motion sequence is known, using these two methods to reconstruct the motion sequence will cause a large reconstruction error, because they completely discard the constraints of the non-keyframe parts on the rotation information curves.

3 The proposed Methodology

We solved two problems, keyframe extraction and motion data reconstruction, which will be discussed separately in the following sections.

3.1 KeyFrame Extraction Method

The keyframe extraction method consists of three steps: curve simplification, key point and keyframes acquisition, binomial fitting parameters acquisition. We treat the rotation information in each direction of the joints and the movement information in each direction of the root joint as a single curve. Then the binomial fitting is used to fit the curve in segments, and the points of each segment of the curve are obtained. All of these points are used as a set for density clustering, and the density centers are extracted as the keyframes. Finally, the rotation information curves are segmented by the keyframes, and binomial fitting is performed by the least square method to obtain the fitting parameters.

3.1.1 Simplified Curve Model

When human body moves, many joints change at the same time. Therefore, the information is rich and the calculation is complicated. Human motion capture records the rotation information of each joint and the translation information of the root joint. This information is three-dimensional, and each dimension of each joint records a piece of information. Some joints have only one or two degrees of freedom, so the rotation information is only one or two dimensions. According to the characteristics of human motion capture data, we take time as the independent variable and the motion information of each dimension of human joints as dependent variables, as a result of which, the integrated motion is reduced to simple curves. We use the curve L_i to represent the change of rotation of a joint in one direction or the trajectory of the root joint in one direction. For each motion sequence, we can obtain 55 rotation information curves and 3 motion trajectory curves. Then the motion sequence is expressed as a set of curves $M = L_1, \dots, L_i, \dots, L_m$, where i represents the i th curve and m is 58.

3.1.2 Key Points Extraction

It is difficult to describe the continuous rotation information curves with the function $L = f(x)$, but it is feasible to divide the curve into small segments and then use binomial fitting. So L_i

can be expressed as:

$$L_i = \begin{cases} l_{i,1} = f_{i,1}(x_j) & j \in [1, n_{i1}) \\ l_{i,2} = f_{i,2}(x_j) & j \in [n_{i1}, n_{i2}) \\ \vdots & \\ l_{i,k} = f_{i,k}(x_j) & j \in [n_{ik-1}, n_{ik}) \end{cases} \quad (1)$$

Where n_1, n_2 and n_{k-1} are segment points and there are n frames in the motion sequence, and j means the j th frame. So the rotation information curves is divided into k segments to fit. For different L_i , the value of k is also different. We want to keep the fitting effect and the fewer segments, the better. In order to achieve this goal, the test index of goodness-of-fit of regression equation R^2 is introduced to restrict the curve piecewise fitting and find the longest piecewise fitting. R^2 is calculated as follows:

$$R^2 = \frac{SSR}{SST} = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1 - \frac{SSE}{SST} \quad (2)$$

$$= 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Where SST is the total sum of squares, SSR is the sum of regression squares, SSE is the sum of residual squares, and $SST = SSR + SSE$, and $SST = \sum_{i=1}^n (y_i - \bar{y})^2$, $SSR = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$, $SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2$.

We designed a self-reducing step length based on the idea of dichotomy to get the segmented points of the rotation information curves according to Algorithm 1. The specific producer is summarized as Algorithm 1: Where $R^2(start, old_end)$ represents the goodness-of-fit between $start$ and old_end of each section. R^2 represents the goodness-of-fit specified by the user. *Keylist* stores the segment points obtained. Using Algorithm 1, we can minimize the number of segments of the rotation information curves, and each segment is the longest, under the current R^2 .

We used binomial $f(x) = a_0 + a_1x$ to fitting the rotation information curves because it is sensitive to changes in the slope of a curve. Points taken from the parts of the curves where the slope changes dramatically can better indicate the

Algorithm 1 Get the segment points

```
1:  $Keylist(1) = 1, i = 1$ 
2: Input  $R^2$ 
3: Set step size to  $step$ , starting point  $start = 1$ , temporary ending point  $old\_end = start + step$ , final ending point  $end = start$ 
4: while  $start \neq n$  do
5:   while  $old\_end \neq end$  do
6:     if  $R^2(start, old\_end) \geq R^2$  then
7:        $end = old\_end$ 
8:        $old\_end = old\_end + step$ 
9:     else
10:       $step = |old\_end - end|/2$ 
11:       $end = old\_end$ 
12:       $old\_end = old\_end - step$ 
13:    end if
14:  end while
15: end while
16:  $Keylist(i++) = end$ 
```

trend of the curves. In order to ensure that the fitting result meets the given goodness of fit R^2 , the curve will be segmented in a timely manner when a region with a large change in slope is encountered. Therefore, binomial fitting can be used to obtain the key points that can represent the overall change of the curves without calculating the curvature change of the curves.

3.1.3 Determining KeyFrame and Obtaining Binomial Parameter

Although the segment points of each curve are different, they are all taken from the continuous frame numbers between 1 and n (suppose the motion sequence has n frames), so some frames must be selected multiple times as segmentation points. For example, assuming that m curves are piecewise fitted, according to Algorithm 1, the first frame and the last frame would be selected as segment points m times. For the same reason, other frames may also be selected as segment points multiple times. As shown in Figure 1, some frames are selected as segmented points many times. But other frames near these frames are selected as segmented points fewer times. It indicates that the segment points are in a clustered state, so we can select keyframes from the clusters. Obviously, the density centers are the best candidate keyframes if density clustering is carried out on the set of segment points. In

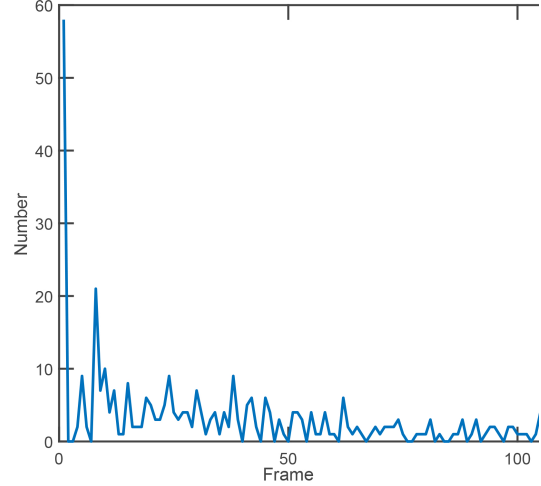


Figure 1: Number distribution of key points

Figure 1, the horizontal axis represents the frame ordinal number, and the vertical axis represents the number of times the frame has been recorded

According to the characteristics of the segment point set, we chose DBSCAN (Density-Based Spatial Clustering of Applications with Noise) for clustering. ε neighborhood and MinPts are the most important impact factors of DBSCAN clustering, which we will discuss in the section 4.

Another thing that needs special mention is that in this clustering algorithm, the samples of the segment points set we used as input are not the coordinates of points in multidimensional space, but the coordinates on one-dimensional space. They are the serial frame numbers of the motion sequence.

Using DBSCAN to cluster segment point set, clustering results $Set1, \dots, Setk, \dots$ and $Setp$ are obtained. The keyframes sequence $keyframelist$ is obtained by taking cluster center $center_k$ as the keyframe. The cluster center is the point with the smallest sum of distance from all other points. For the i th point in the class $Setk$, the sum of the distances of the other points from it can be expressed as:

$$Dist(i_k) = \sum_{j_k=FF_k}^{LF_k} (j_k - i_k)N_j \quad (3)$$

$i_k, j_k \in Setk$ is the serial frame number, FF_k is the first frame in $Setk$, LF_k is the last frame in $Setk$, and N_j means that the j_k th frame was selected as a segment point N_j times. So the cluster center $center_k$ satisfies:

$$Dist(center_k) = \min(Dist(i_k)) \quad (4)$$

where $i_k \in \text{Set}k$.

Finally, we need to add the first frame 1 and the last frame n of the motion sequence to the *keyframelist* if they don't exist. At the same time, if there are consecutive frames in the *keyList*, we should delete some of them and keep only one of them.

3.2 Motion data Reconstruction Method

We proposed a novel method for motion data recovery based on binomial coefficients. This method is different from quaternion spherical interpolation algorithm and linear interpolation algorithm. In addition to the keyframe information, the binomial coefficients need to be obtained in advance. After getting the keyframe sequence, we still need to get the binomial coefficients. Figure 2 shows the process of getting the parameters.

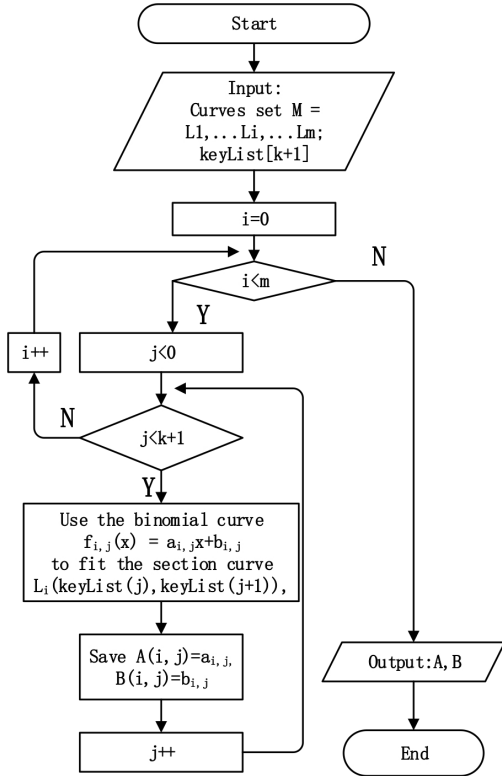


Figure 2: Clustering effects

So the motion curve set M can be reconstructed through the keyframes and binomial coefficients set A and B . As for the rotation informa-

tion curve L_i , it can be reconstructed as:

$$L_i = \begin{cases} f_{i,1}(x) = a_{i,1}x + b_{i,1} & x \in [kL(1), kL(2)) \\ \vdots \\ f_{i,j}(x) = a_{i,j}x + b_{i,j} & x \in [kL(j), kL(j+1)) \\ \vdots \\ f_{i,k}(x) = a_{i,k}x + b_{i,k} & x \in [kL(k), kL(end)] \end{cases} \quad (5)$$

j represents the j th segment of curve L_i , which is divided into k segments by keyframes, kL represents *keyframelist*.

4 Discussion of parameters

There are three parameters need to be determined in the proposed method: the distance threshold ε , the neighborhood sample threshold MinPts and the goodness-of-fit R^2 . We tested 200 exercise videos randomly selected from the Carnegie Mellon University (CMU) Motion Capture Database. In this paper, we introduced the mean error of joint space coordinates as the evaluation metric illustrate the quality of the keyframe extraction results. The error E is calculated as follows:

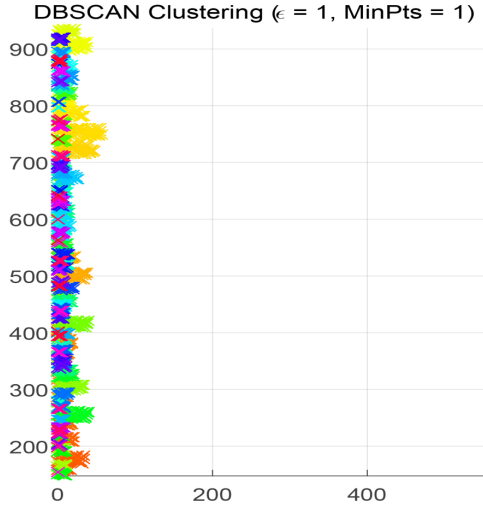
$$E = \frac{\sum_{i=1}^s \sum_{j=1}^n \|F_{i,j}^o - F_{i,j}^r\|_2}{s \times n} \quad (6)$$

In order to compare with similar papers, the reconstruction error is calculated based on the trajectory curve of the joint. So we first use the rotation information of the joint to calculate the trajectory curve and then calculate the reconstruction error. So in this formula, $F_{i,j}^o$ is the j th joint space coordinate in the i th frame of the original motion sequence, and $F_{i,j}^r$ is the j th joint space coordinate in the i th frame of the reconstruction motion sequence. s is the number of joints, n is the number of frames.

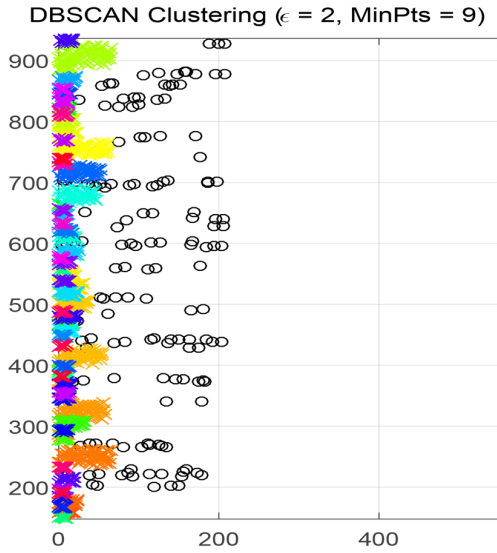
4.1 ε and MinPts

ε represents the distance between samples. For the reason that samples themselves are integers, the distance between the samples is also an integer, so the best value of ε is an integer.

In our experiments, when ε is larger than 1, outliers will increase, and the reconstruction error of extracted keyframes is much larger than that of ε is equal to 1, so the clustering effect is poor, as shown in Figure 3(b). If ε is 0, every frame that appears will be individually clustered into a cluster, which leads to serious redundancy of keyframes, so it is not suitable either. But as the Figure 3(a) shows, when $\varepsilon=1$ and MinPts



(a) $\varepsilon=1$ MinPt=1



(b) $\varepsilon=2$ MinPt=12

Figure 3: Clustering effects

takes the appropriate value, the clustering effect is decent.

Figure 3(a) shows the best clustering effect of a motion sequence when ε is equal to 1 and Figure 3(b) shows the best clustering effect of a motion sequence when ε is equal to 2. The x-coordinate represents the number of elements in the same class, and the y-coordinate represents the frame number. \circ indicates outliers, and the same color of x represents one cluster. It can be seen that there are a large number of outliers in Figure 3(b). This indicates that the clustering effect is the best when ε is equal to 1.

MinPts is the minimum number of neighborhood samples. In DBSCAN, its difficult to select MinPts. In order to find the optimal value range of MinPts, we conducted such an experiment: ε

is 1, and R^2 is set at 0.05 intervals between 0.05 and 0.90. When R^2 is less than 0.90, the reconstruction error of all motion sequences tested is less than 0.20, and the average reconstruction error is less than 0.03. When R^2 is greater than 0.90, the reconstruction error of some motion sequences is above 3.0, and some are even as high as 63.45, and the average reconstruction error is also above 1.00. Therefore, when R^2 is greater than 0.90, the reconstruction effect is not only poor, but also unstable. So in this article, we will only discuss the part where R^2 is small than 0.90. For each R^2 value, MinPts starts at 1 and increments by 1 for each cycle. In each loop, we obtained the keyframes of the input motion sequence and calculated the RICE (the reconstruction error of rotation information curves).

By comparing RICE corresponding to each MinPts, select the MinPts with the minimum RICE, which is the best MinPts corresponding to the current R^2 value. Under the condition that ε is 1, we conducted experiments on 200 motion sequences and obtained the optimal MinPts range for each R^2 value, as shown in Table 1.

4.2 Coefficient of Determination R^2

R^2 has a significant effect on keyframe extraction. The larger the R^2 is, the more segments the rotation information curve will segment. However, the more segmentation points there are, the closer the density connection between samples will be, and some important frames will be classified as one category due to the increase of the density, leading to a result that only one of several potential keyframes is ultimately extracted, which leads to insufficient extraction of keyframes and the increase of motion data reconstruction error. We tested 200 motion sequences to explore the effect of R^2 on compression ratio (the ratio of the number of keyframes to the total number of frames) and reconstruction error. Figure 4 and Figure 5 reflect the influence of R^2 on the compression ratio and the compression ratio on the reconstruction error.

Figure 4 shows the relationship between R^2 and compression ratio. As can be seen from Figure 4, compression increases when R^2 between 0 and 0.3, after that it decreases. Figure 5 reflects the relationship between the compression rate and the reconstruction error. In general, the

Table 1: Different MinPts ranges with different values of R^2

R^2	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
MinPts	1	1	1	1	1	1	1	1	1-5	1	1-5	1-8	1-7	1-10	1-13	1-15	6-18	7-23

reconstruction error is small than 0.16. As the compression rate increases, the reconstruction error gradually decreases. When compression ratio is small then 0.1537, the error decreases rapidly. When compression ratio is larger then 0.1537, the error decreases slowly. For keyframe extraction attempts to ensure a small compression rate and reconstruction error at the same time, the R^2 corresponding to a compression rate of 0.1537 is the most appropriate, with a value of 0.65, as shown in Figure 4.

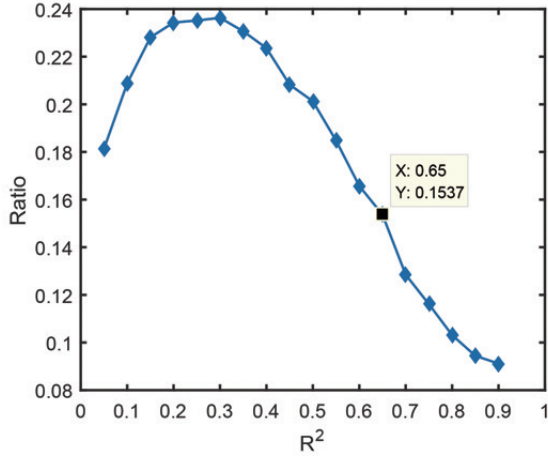


Figure 4: Compression Ratio and R^2

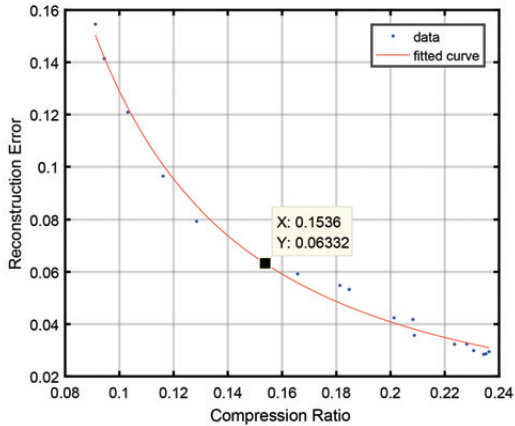


Figure 5: Reconstruction Error and Compression Ratio

Table 2: Results of Keyframe Extraction

classes	jump	cartwheel	dance
Total Frames	295	575	1131
Number of Keyframes	37	69	134
Compression ratio	12.50%	12.00%	11.80%
Computation time(s)	9.42	15.46	41.95

5 Result

In section 4, we determined that parameter ε is 1, R^2 is 0.65, and R^2 corresponds to MinPts ranging from 1 to 7. In this section, we will use these parameters to perform keyframe extraction and reconstruction error calculation, and compare the advantages and disadvantages of our motion data reconstruction algorithm and the traditional motion data reconstruction algorithm, as well as the advantages and disadvantages of our keyframe extraction algorithm and the traditional keyframe extraction algorithm.

Figure 6, show the key frames extracted from different motions. As can be seen from the figure, the key frame extracted by our method can well represent the changing trend of the motions.

Table 2 lists the number of keyframes, compression ratios, and calculation time for different action classes. As can be seen from the results, the compression ratio and computation time give an ideal result.

The keyframes extracted by our method are reconstructed using three different motion data reconstruction methods. Table 3 lists the reconstruction error of three different reconstruction methods for the three classes motions. BFE is the error of the motion data reconstruction algorithm we proposed, LIE is the reconstruction error of the linear interpolation algorithm, and QSIE is the reconstruction error of the quaternion spherical interpolation algorithm. As can be seen from the results that the reconstruction effect of linear interpolation is the worst, followed by QSIE, and the motion data reconstruction algorithm proposed by us has the best effect.

It is easy to understand that our method is better than linear interpolation. The linear interpolation formula is shown below:

$$f_{lerp}(p, q, t) = t(p - q) + p \quad (7)$$

Table 3: Results of different reconstruction methods

classes	jump	cartwheel	dance
BFE	0.0095	0.0793	0.0095
LIE	0.0599	0.2094	0.0176
QSIE	0.0218	0.0176	0.0172

p and q are the values of two adjacent keyframes in the keyframe sequence and represent the ordinate of the original curve when reconstructing rotation information curve, and t is the interval. In fact, two adjacent keyframes are connected by line segments, and then intermediate frames are found on the line segments according to the interval. Our method fits a curve with the smallest regression error according to the original rotation information curve in advance, and reconstructing the data is to restore the fitting curve.

Quaternion can reconstruct smooth motion trajectories, but they also have limitations. The condition for applying the quaternion is to convert the rotation information of the joint into a quaternion expression. Its algorithm is as follows:

$$f_{slerp}(P, Q, t) = \frac{\sin[(1-t)\theta]P + \sin t\theta Q}{\sin\theta} \quad (8)$$

P , Q represents a vector containing the three-dimensional rotation information of the joint, while the rotation information curve represents one-dimensional information, and the three-dimensional information is represented by three pieces of curves. θ is the angle between the vector P and Q . So the points quaternion spherical interpolation obtained is on the rotation trajectory of vector P , which is on the plane S determined by P and Q . However, the vector P may rotate from the space outside the S to the Q . For example, when we straighten our arms, the fist has more than one path from point A to point B in space, but the quaternion spherical interpolation algorithm can only calculate the trace on the plane determined by points A , B and shoulder joint C . Our method reconstructs the rotation information of each dimension of the joint, so the trajectory found by our method is closer to the real motion path than the trajectory calculated by the quaternion spherical interpolation algorithm, and the error is smaller.

Compared with the quaternion spherical interpolation and linear interpolation algorithms, the binomial fitting based reconstruction algorithm has smaller reconstruction error, but it needs

extra storage space to store parameters. When using quaternion spherical interpolation algorithm or linear interpolation algorithm to reconstruct the motion sequence, not only the keyframe sequence but also the motion information corresponding to the keyframe is needed. Assuming that a motion sequence with n frames can decompose m motion curves, the total motion information of the sequence is $m * n$. If k keyframes are extracted from them, the total amount of information to be recorded is $k + k * m$ for motion reconstruction. Therefore, after extracting the key, the ratio of the stored information and the total amount of the original motion information is $(k + k * m) / (n * m) = k / n$, namely the compression ratio of the keyframe. If binomial fitting reconstruction is used, according to $y = ax + b$, although the motion information corresponding to the key frame is not recorded, fitting parameters should be recorded. According to Figure 2, each curve segment needs to record two parameter information, so the total motion information to be recorded is $k + 2 * k * m$. The ratio of stored information to the total amount of original motion information is $(k + 2 * k * m) / (n * m) = 2 * k / n$. Therefore, using binomial fitting to reconstruct the motion sequence requires about twice as much information to be saved when extracting the key frame as the other two methods.

Figure 7, 8, 9 show the comparison between the original sequence and the reconstructed sequence based on the keyframes extracted by our method, joint kernel sparse representation algorithm and affinity propagation algorithm, where green is the original data and red is the reconstructed result.

In Figure 7 dance motion sequence has 1131 frames, and the number of keyframes is 134 with compression ratio of 11.80%. As can be seen from Figure 7(a), there is little difference between the reconstructed frames and the original data. Figure 7(b) shows a significant difference between the reconstructed results and the original data at the foots after dance motion beginning. In Figure 7(c), the errors at foots and head are obvious at the beginning, and later the upper body also appeared a slight error.

In Figure 8 dance motion sequence has 575 frames, and the number of keyframes is 69 with compression ratio of 12.0%. It can be seen from Figure 8(a) that there is no obvious error in the reconstruction results except that a little error in

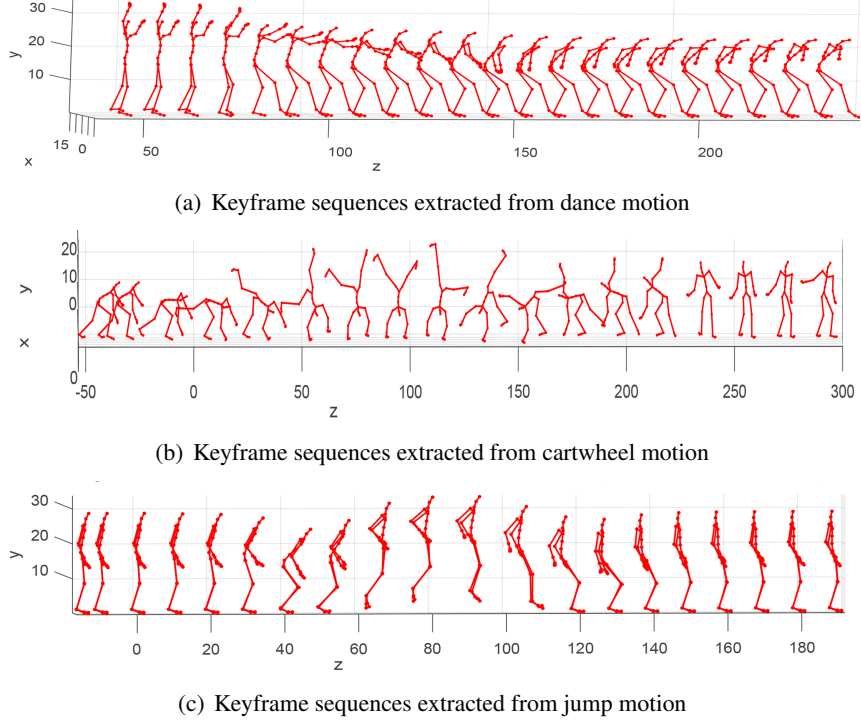


Figure 6: Results of keyframe extraction

the ball of the foot at the beginning of the motion. However, it can be seen from Figure 8(b) and Figure 8(c) that there is a large error between the reconstruction results and the original data. In Figure 8(b), the error is significant at the beginning phases of the motion and in the middle phases of the motion, the reconstructed results are markedly different from the original data. In Figure 8(c), there is a noticeable error at the beginning of the motion, and before the end of the motion, the difference between the reconstructed results and the original data is obvious.

In Figure 9 jump motion sequence has 295 frames, and the number of keyframes is 37 with compression ratio of 12.5%, Figure 9(a) shows a good reconstruction effect, with only slight errors in the feet. In Figure 9(b), there is a significant error in the arm when the person jumps. In Figure 9(c), there are errors in both the arms and feet during the squat and the jump.

Table 4 compares the reconstruction errors of the three keyframe extraction algorithms. AE represents Affine Propagation algorithm[12] and JE represents Joint Kernel Representation algorithm[16]. Since both algorithms use quaternion method for reconstruction, we compare the reconstruction error of using quaternion method with them. As can be seen from the experimental results, the reconstruction error of the proposed

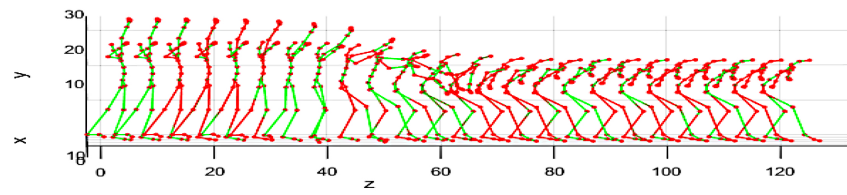
Table 4: Reconstruction error of three keyframe extraction algorithms

classes	jump	cartwheel	dance
Our method	0.0218	0.0176	0.0172
AE	0.4518	0.2566	0.248
JE	0.1483	0.47	0.0324

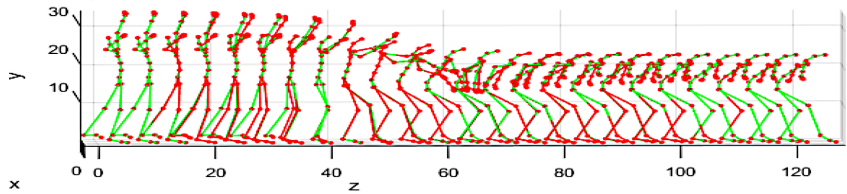
method is significantly better than Affine Propagation algorithm and Joint Kernel Representation algorithm.

6 Conclusion

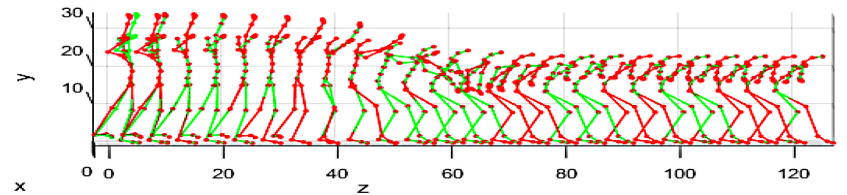
We proposed a new keyframe extraction algorithm and a motion data reconstruction method based on multiple binomial fitting. They are simple and effective. These two algorithms comprehensively consider the motion changes of various parts of the joint and obtain impressive results. Experiments show that the proposed method is superior to the linear interpolation algorithm and the quaternion spherical interpolation algorithm. Although the key frame compression rate extracted by our algorithm is relatively large, averaging over 10%, which needs to be improved in future work. But the reconstruction error is much smaller than the algorithm based on clustering and optimization.



(a) The proposed method

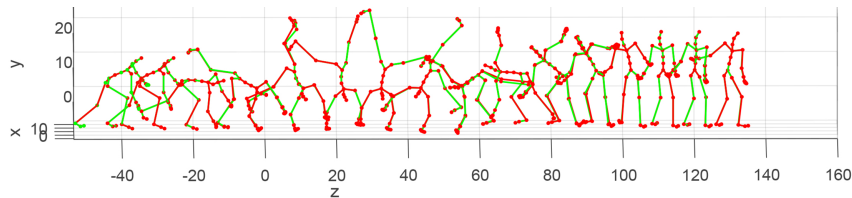


(b) Joint kernel sparse representation algorithm

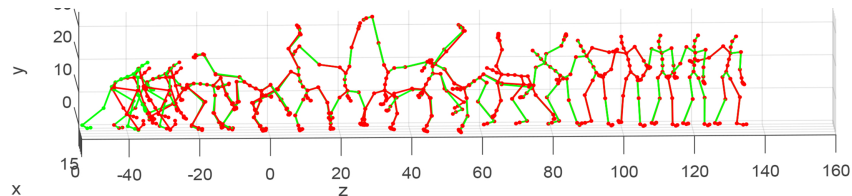


(c) Affinity propagation algorithm

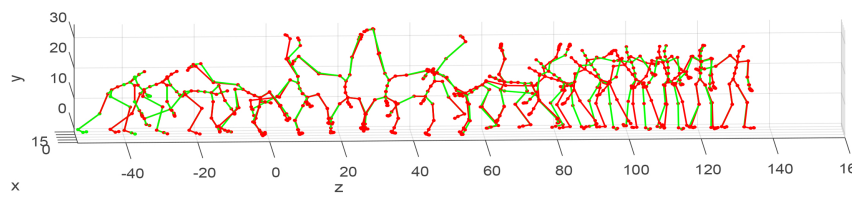
Figure 7: Results of original dance motion and reconstructed motion



(a) The proposed method



(b) Joint kernel sparse representation algorithm



(c) Affinity propagation algorithm

Figure 8: Results of original cartwheel motion and reconstructed motion

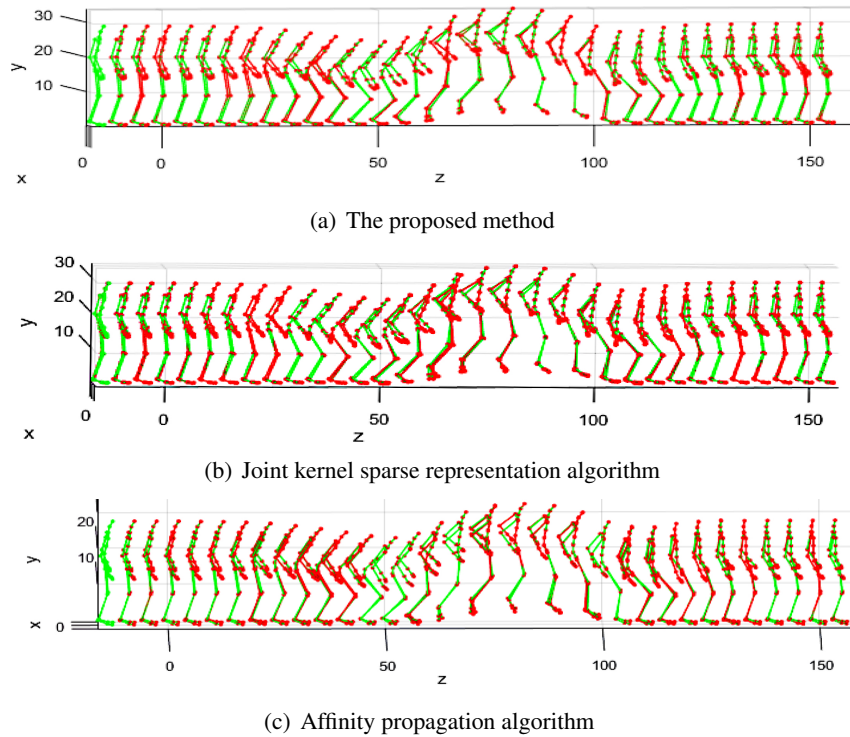


Figure 9: Results of original jump motion and reconstructed motion

Acknowledgement

This work is partially funded by Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, China(2018AIOT-09), Key Research and Development Program of Shaanxi Province(2018NY-127), and the Shaanxi Key Industrial Innovation Chain Project in Agricultural Domain(Grant No. 2019ZDLNY02-05).The authors acknowledge Carnegie Mellon University for the motion capture data resources.

References

- [1] Z. J. Li, C. Y. Su, L. Y. Wang, et al. Non-linear Disturbance Observer-Based Control Design for a Robotic Exoskeleton Incorporating Fuzzy Approximation. *Ieee Transactions on Industrial Electronics*, 62(9):5763-5775, 2015.
- [2] S. Y. Shin, C. Kim Human-Like Motion Generation and Control for Humanoid's Dual Arm Object Manipulation. *Ieee Transactions on Industrial Electronics*, 62(4):2265-2276, 2015.
- [3] T. Sasaki, D. Brscic, H. Hashimoto Human-Observation-Based Extraction of Path Patterns for Mobile Robot Navigation. *Ieee Transactions on Industrial Electronics*, 57(4):1401-1410, 2010.
- [4] Y. R. Li, Z. Wang, X. S. Yang, et al. Efficient convolutional hierarchical autoencoder for human motion prediction. *Visual Comput*, 35(6-8):1143-1156, 2019.
- [5] C. Halit, Capin T. Multiscale motion saliency for keyframe extraction from motion capture sequences. *Comput Animat Virt W*, 22(1):3-14, 2011.
- [6] T. Miura, T. Kaiga, T. Shibata, et al. A hybrid approach to keyframe extraction from motion capture data using curve simplification and principal component analysis[J]. *IEEJ Transactions on Electrical and Electronic Engineering*, 2015, 9(6):697-699.
- [7] Jun Xiao, Yueting Zhuang, Tao Yang, et al. An Efficient Keyframe Extraction from Motion Capture Data[C]// *International Conference on Advances in Computer Graphics*. Springer-Verlag, 2006.
- [8] K. S. Huang, C. F. Chang, Y. Y. Hsu, et al. Key Probe: a technique for animation keyframe extraction. *Visual Comput*, 21(8-10):532-541, 2005.

- [9] Y. H. Gong, X. Liu Video summarization and retrieval using singular value decomposition. *Multimedia Syst*, 9(2):157-168, 2003.
- [10] X. M. Liu, A. M. Hao, D. Zhao Optimization-based keyframe extraction for motion capture animation. *Visual Comput*, 29(1):85-95, 2013.
- [11] X. j. Chang, P. F. Yi et al. Key Frames Extraction from Human Motion Capture Data Based on Hybrid Particle Swarm Optimization Algorithm[J], *Recent Developments in Intelligent Information and Database Systems*, pp335-342.
- [12] B. Sun, Kong D., S. Wang, et al. Keyframe extraction for human motion capture data based on affinity propagation[C]// 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE, 2018.
- [13] Q. Zhang, S. P. Yu, D. Zhou, et al. An Efficient Method of Key-Frame Extraction Based on a Cluster Algorithm. *J Hum Kinet*, 39(1):5-13, 2013.
- [14] Y. Zhang, J. Li, M. Zhang, et al. Motion Key frame Extraction Based on Grey Wolf Optimization Algorithm[J]. *MATEC Web of Conferences*, 2018, 232.
- [15] Q. Zhang, S. Zhang, D. Zhou, et al. Keyframe Extraction from Human Motion Capture Data Based on a Multiple Population Genetic Algorithm[J]. *Symmetry*, 2014, 6(4):926-937.
- [16] G. Y. Xia, H. J. Sun, X. Q. Niu, et al. Keyframe Extraction for Human Motion Capture Data Based on Joint Kernel Sparse Representation. *Ieee Transactions on Industrial Electronics*, 64(2):1589-1599, 2017.
- [17] Q. Z, G. Z, L. X, et al. Human Motion Capture Error Correction Method without Force-Feedback[C]. *international conference on artificial intelligence*, 2019.